Mastering Embedded Systems

194 Questions – ECE 332 Exam Preparation

(also useful in preparing for job interviews)

Grad TAs: Sachin Bhat, Chandra Sekhar

Note: We put these together to help you prepare and test your knowledge. The questions follow the lectures and labs we have completed and try to capture the most foundational aspects of embedded system design. Answers should be easily accessible from the slides, notes. Any questions please use piazza.

Questions*:*

1. What is an embedded system?
2. What are some design constraints usually involved in embedded system design?
3. Mention few characteristics that are typical of embedded systems.
4. How do embedded systems differ from general purpose computers?
5. Give some real-world examples of embedded systems.
6. Mention the two types of time constraints during execution in designing embedded systems
7. What is a microprocessor?
8. List few embedded microprocessors, their data width and vendors.
9. What is a microcontroller?
10. Give example of microcontrollers. List differences vs microprocessors.
11. What is an FPGA?
12. List the differences between a hard-core and soft-core processor?
13. What kind of processors can be supported in an FPGA?
14. List few embedded microprocessor architectures.
15. Explain the von Neumann architecture in brief. What is it's characteristic feature?
16. Explain the Harvard processor architecture organization.
17. What is an instruction set? Why is it needed? Hos is it different than CPU assembler language?
18. Expand CISC.
19. What is a CISC architecture? What are its defining characteristics? Give an example.
20. Expand RISC.
21. What is a RISC architecture? What are its defining characteristics? Give an example.
22. Would you prefer to use CISC or RISC for embedded applications? Why?
23. Explain pipelining. Could you give a real-world example of pipelining in a processor like ARM?
24. What are the typical stages of a 3-stage pipeline? What about a 5-stage pipeline?
25. What would take more resources for a given problem: implementing custom logic or using a microprocessor in silicon for the same problem? Elaborate.

26. What aspects of microprocessors make it better than custom logic in some cases? What aspects make it worse?
27. How is a soft-core processor implemented in an FPGA? What is a lookup table?
28. Rank performance between FPGA softcore and hardcore microprocessor?
29. Discuss power consumption tradeoffs in implementations of RLE between ASIC hardware, FPGA hardware, softcore processor software, software on hardcore processor?
30. Explain a scenario where you would prefer to use soft-core processor over a hard-core processor.
31. Is NIOS a RISC or CISC processor? Give two reasons to support your answer.
32. What is the name of the softcore processor which you worked with in Lab 1?
33. Who is the vendor? Also, include another vendor of SRAM based FPGAs.
34. Expand SOPC and explain its significance for FPGA based hardware design.
35. What are the two flavors of NIOS II that you could instantiate on the DE1-SoC board?
36. What purpose does the JTAG UART serve in the QSYS of Lab 1? What is JTAG?
37. What is pin assignment and why is it necessary? How do you connect between inner signals in an FPGA and outside components.
38. What is on-chip memory? How is on-chip memory implemented (technology) on the FPGA?
39. What is the name of the logic interconnect bus used in the QSYS to connect components?
40. What is SDRAM? Is it volatile or non-volatile? Is SDRAM faster or slower than SRAM?
41. Is the SDRAM on the same chip as the FPGA chip?
42. Why is SDRAM required for the part 2 of the lab?
43. Is a processor cache SRAM or SDRAM based?
44. Why is the instruction master (fetch unit) of NIOS II processor only connected to SDRAM controller component and not to any other component (in part 2)?
45. Why did you need to use a PLL component in the part 2 of Lab 1?
46. What are IRQs?
47. What is difference between level-triggered IRQ and edge-triggered IRQ? Which type did you use in part 2 of the lab?
48. What is the outcome after you ‘generate’ the QSYS system? What is the type of file generated and what does it contain?
49. Why is SDRAM controller required in the lab? Is it a hard IP or a soft IP?
50. Why was the top-level Verilog file required for part 2 of lab 1?
51. Explain how you did the conversion from hexadecimal to decimal in the count binary program.
52. List the differences between RISC (scalar) vs Superscalar. Give few examples for each.
53. What is out-of-order execution? What are the benefits vs in-order execution?
54. What is a compiler?
55. What are VLIW architectures? Give an example.
56. What kind of parallelism does VLIW architectures exploit?
57. List key differences between microcontrollers, microprocessors and FPGA systems.
58. Expand FPGA.
59. List key differences between ASICs and FPGAs.
60. List the components in memory system of a typical CPU.

61. Show a design(block diagram) of an embedded memory system with TLB and page table.
62. What is memory hierarchy? Why is it required?
63. What are typical access times in CPU clock cycles for L1 cache, L2 cache and main memory?
64. What are the types of computer memory? Include persistent and volatile memory.
65. What is the memory type used to implement caches? What about to store firmware?
66. What is the memory type used to main external memory?
67. Draw a typical memory device organization.
68. What is a memory management unit (MMU)?
69. Show a design of an embedded memory system with a relocation register.
70. What is the disadvantage of the relocation register-based system?
71. What is virtual memory?
72. Draw the virtual memory organization.
73. What is a TLB?
74. What is a page table?
75. What is a logical address?
76. What is a physical address?
77. What is assembly language?
78. What is CPU machine code?
79. What are high-level programming languages?
80. Draw a typical interrupt interface.
81. List the differences between maskable and non-maskable interrupts.
82. Draw the generalized interrupt mechanism.
83. What is an interrupt vector table?
84. List the sources of interrupt overhead.
85. With a flowchart, explain the generic interrupt mechanism.
86. With a diagram, explain how prioritized I/O would work.
87. Why might using nested function calls within an interrupt routine be problematic?
88. How would you prioritize the interrupts for the following? (a) UART (b) JTAG (c) DN9000.
89. List the differences between hardware and software interrupts.
90. List the differences between exception and trap.
91. What is serial communication?
92. Expand UART.
93. Explain the generic asynchronous serial communication protocol.
94. List all the serial communication parameters.
95. What are the two types of instructions that can support I/O programming?
96. What is meant by memory-mapped I/O? How is that vs. IO instructions?
97. Expand USB.
98. Explain how the USB protocol works.
99. What is PCIe architecture? Is it bus-based or point-to-point?
100. Describe how the PCIe architecture works briefly.
101. What is firmware?
102. What is a device driver?

103. Mention the steps involved in building a code image.
104. Mention steps involved in building an FPGA bitstream (bitstream = content to be loaded into FPGA).
105. Mention steps involved in building an ASIC. How is it different from an FPGA flow?
106. What are assemblers?
107. What is linking?
108. What is dynamic linking?
109. List the differences static and dynamic linking.
110. Why are dynamically linked libraries(DLLs) preferred compared to statically linked libraries in embedded systems?
111. When you use a #include of a .h file, what are you doing? Name one .h file you used in the labs.
112. What are state-machines?
113. What are in-circuit emulators?
114. Write the pseudocode for implementing state machine using switch and case statements.
115. How do you debug a target system with a host system?
116. What is a pixel?
117. What is a picture?
118. What is image resolution?
119. What is a grayscale image?
120. What is quantization? Why is it necessary?
121. Why is image compression necessary?
122. Explain the trade-off between quality and picture storage requirements.
123. How are images represented digitally?
124. What is the resolution of the image being displayed on the screen in Lab 2?
125. What processor was used to perform image processing in this lab?
126. What is a DMA controller?
127. What are the two DMA controllers used in lab 2? What are they used for?
128. What is a VGA controller? What is it used for?
129. How does the processor access the states of buttons and switches?
130. What component displays the image on to the screen?
131. Why do we need to downsample the image before storing it? What hardware components are used to perform this operation?
132. Where are the images stored?
133. How did you implement mirroring operation?
134. How did you implement flipping operation?
135. How did you implement conversion to black and white?
136. Explain how would you convert a color image to a grayscale image.
137. How is text displayed on the monitor?
138. What purpose does the Altera monitor program serve?
139. What is an operating system?
140. What are the important functions of operating systems?
141. What is a process?

142. How are the two ways a process is created?
143. What are the various states a process is going through during its lifecycle?Show the state machine for process scheduling.
144. What is a process control block?
145. Mention the components of the process control block or PCB.
146. What is a thread?
147. What is a single thread process?
148. What is a multi-threaded process?
149. List the differences between threads and processes.
150. List the differences between embedded and general-purpose scheduling.
151. Illustrate how the CPU switches from process to process with a diagram.
152. What is interprocess communication?
153. List two different styles of interprocess communication. Explain each briefly.
154. Explain the concept of shared memory interprocess communication with a diagram.
155. What is message passing inter-process communication?
156. What are critical regions in sections of code? Give two examples.
157. With an example, explain how a race condition might occur in shared memory.
158. With a pseudocode, explain the mutual exclusion lock with test and set.
159. Please explain why a test-and-set instruction or similar hardware is necessary for enabling proper synchronization.
160. What are semaphores? What are the two types of semaphores?
161. Explain the differences between a counting semaphore and a simple lock.
162. Why is it important that a Test-and-Set function be atomic?
163. Why is it useful to implement some application in hardware (e.g., FPGA) vs implementing in software (processor)?
164. What is Run-Length Encoding? Briefly describe the encoding protocol used in lab 3.
165. What is the length of input and output bit streams in RLE implemented in lab 3?
166. Mention some of the other compression algorithms.
167. What are the differences between lossy and lossless compression algorithms.
168. Explain difference between blocking and non-blocking assignments.
169. When would you use blocking assignments? When would you use non-blocking assignments? What did you use for this lab?
170. Explain difference between lossy and lossless compression. Which type is RLE?
171. What is pattern of bits for which RLE does the worst? What pattern would it do the best?
172. What is the shortest encoding possible (with our Lab setup) with RLE for an image that is all black but has a single white pixel in the middle. What is the size of the RLE encoded image in that case?
173. How many repetitions of 1s or 0s need to be input before the RLE starts becoming beneficial?
174. How would you modify the RLE encoding protocol such that it is more effective than what it is in lab 3?
175. Would the RLE work better for a color image or a BW image?
176. What does hardware verification mean? What is a testbench?

177. What is functional simulation? Why is it necessary?
178. Draw the testbench template.
179. What are the advantages of hardware/software co-design?
180. What is a FIFO buffer?
181. Why is FIFO buffer necessary in the hardware implementation of the RLE encoder?
182. How did the software decompression you wrote in Lab 4 access the output of the hardware RLE encoder?
183. Draw the system diagram of lab 4 setup.
184. How does the ARM HPS access the hardware instantiated on the FPGA fabric?
185. What is the purpose of preprocessing the image in lab 4?
186. Mention the altera functions used to send and receive data between the ARM HPS and the FPGA.
187. Outline the structure of a sequential block in Verilog. Can it contain combinational logic? What is the proper way to code it?
188. What is a functional simulator for Verilog? Which one did you use in the lab?
189. What is a top level module in Verilog?
190. Why do we need to synthesize Verilog to download to FPGA?
191. What is the difference between designing a processor for an FPGA vs. custom silicon?
192. Are the combinational block statements in Verilog blocking or non blocking? Why?
193. List advantages and disadvantages associated with using bare-metal ARM HPS vs Linux running on ARM HPS.
194. What is HPS-FPGA lightweight bridge? How is it used in lab 4?

Acknowledgment: These questions are based on compilation by Sachin Bhat, Sourabh Kulkarni and Prof. Csaba Andras Moritz.